

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平10-161933

(43) 公開日 平成10年(1998) 6月19日

|                            |       |
|----------------------------|-------|
| (51) Int. Cl. <sup>8</sup> | 識別記号  |
| G 0 6 F 12/14              | 3 1 0 |
| 9/46                       | 3 4 0 |

FI  
G06F 12/14 310A  
9/46 340F

審査請求 未請求 請求項の数24 O L (全 10 頁)

(21)出願番号 特願平9-254505

(22)出願日 平成9年(1997)9月19日

(31) 优先权主张番号 08/721145

(32)優先日 1996年9月26日

(33) 優先權主張國 米國 (US)

(71)出願人 390009581

インターナショナル・ビジネス・マシーンズ・コーポレーション

INTERNATIONAL BUSINESS  
MACHINES CORPORATION

アメリカ合衆国10504、ニューヨーク州  
アーモンク (番地なし)

(72)発明者 ドナルド・フレッド・オールト

アメリカ合衆国12538 ニューヨーク州ハ  
イド・パーク ルーズベルト・ロード  
115

(74) 代理人 弁理士 坂口 博 (外1名)

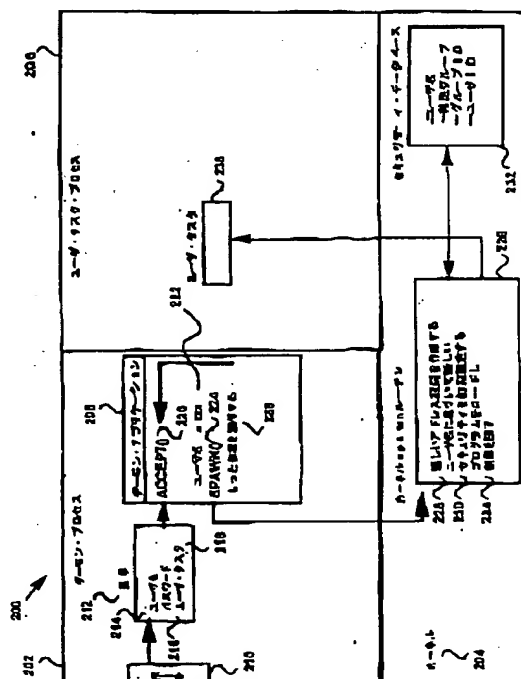
## 最終頁に絞く

(54)【発明の名称】 クライアント／サーバ・システムにおける適切なセキュリティ環境でのユーザ・タスク実行方法及び装置

(57) 【要約】

【課題】 クライアント／サーバシステムにおいて、監視デーモンが指定されたタスクをユーザに代わって実行することができるようにする方法及び装置を提供する。

【解決手段】 監視デーモンは、ユーザ要求を受け取ると、要求で指定されているユーザ・アイデンティティに従って環境変数を設定し、オペレーティング・システム・カーネルに対してシステム・コールを出して要求で指定されているユーザ・タスクを作成する。オペレーティング・システム・カーネルは、このシステム・コールに回答して指定されたユーザ・タスクのための新しいアドレス空間を作成し、環境変数に従ってユーザ・タスクのためのセキュリティ環境を作成してからユーザ・タスクを新しいアドレス空間で開始する。



## 【特許請求の範囲】

【請求項1】 要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視するサーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する方法であって、

(a) デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、[1] 前記要求で指定された前記アイデンティティに従って環境変数を設定するステップと、

[2] 前記オペレーティング・システム・カーネルに対してシステム・コールを出して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

(b) 前記オペレーティング・システム・カーネルが前記デーモン・プロセスから前記システム・コールを受け取ると、[1] 前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、[2] 前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、[3] 前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始するステップとを含む方法。

【請求項2】 前記オペレーティング・システム・カーネルがPOSIX準拠オペレーティング・システム・カーネルであることを特徴とする、請求項1に記載の方法。

【請求項3】 前記システム・コールがspawn()システム・コールであることを特徴とする、請求項1に記載の方法。

【請求項4】 前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、請求項1に記載の方法。

【請求項5】 前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、請求項4に記載の方法。

【請求項6】 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザIDを有し、前記セキュリティ環境を作成する前記ステップ(b)

[2] が、前記新しいユーザ・アドレス空間のユーザIDを、前記環境変数によって指定されたユーザ名のユーザIDと等しく設定することを特徴とする、請求項5に記載の方法。

【請求項7】 前記ユーザ名のユーザIDを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、請求項6に記載の方法。

【請求項8】 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループIDを有し、前記セキュリティ環境を作成する前記ステップ(b)

[2] が、前記新しいアドレス空間のグループIDを、前記環境変数によって指定されたユーザ名のグループIDと等しく設定することを特徴とする、請求項5に記載の方法。

【請求項9】 要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視する前記サーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する装置であって、

(a) 前記デーモン・プロセスに関連づけられ、前記ユーザからの前記要求の受取りに应答して、[1] 前記要求に指定された前記アイデンティティに従って環境変数を設定し、[2] 前記オペレーティング・システム・カーネルに対してシステム・コールを出して新しいアドレス空間内で前記指定されたユーザ・タスクを実行する手段と、

(b) 前記オペレーティング・システム・カーネルに関連づけられ、前記デーモン・プロセスからの前記システム・コールの受取りに应答して、[1] 前記指定されたユーザ・タスクのために新しいアドレス空間を作成し、

[2] 前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成し、[3] 前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始する手段とを含む装置。

【請求項10】 前記オペレーティング・システム・カーネルがPOSIX準拠オペレーティング・システム・カーネルであることを特徴とする、請求項9に記載の装置。

【請求項11】 前記システム・コールがspawn()システム・コールであることを特徴とする、請求項9に記載の装置。

【請求項12】 前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、請求項9に記載の装置。

【請求項13】 前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、請求項12に記載の装置。

【請求項14】 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザIDを有し、前記セキュリティ環境を作成する前記手段(b) [2] が、前記新しいアドレス空間のユーザIDを、前記環境変数によって指定されたユーザ名のユーザIDに等しく設定する手段を含むことを特徴とする、請求項13に記載の装置。

【請求項15】 前記ユーザ名のユーザIDを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、請求項14に記載の装置。

【請求項16】 前記アドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループIDを有し、前記セキュリティ環境を作成する前記ステップ(b)

[2] が、前記新しいアドレス空間のグループIDを、前記環境変数によって指定されたユーザ名のグループIDに等しく設定することを特徴とする、請求項13に記載の装置。

載の装置。

【請求項17】要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・システム・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視する前記サーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わってタスクを実行する方法ステップを行う機械によって実行可能なプログラムを有形に実施する機械可読プログラム記憶装置であって、前記方法ステップが、

(a) 前記デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、[1] 前記要求に指定された前記アイデンティティに従って環境変数を設定するステップと、[2] 前記オペレーティング・システムに対してシステム・コールを発行して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

(b) 前記オペレーティング・システム・カーネルが、前記デーモン・プロセスから前記システム・コールを受け取ると、[1] 前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、[2] 前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、[3] 前記指定されたユーザ・タスクを前記新しいアドレス空間内で開始するステップとを含む、プログラム記憶装置。

【請求項18】前記オペレーティング・システム・カーネルがPOSIX準拠オペレーティング・システム・カーネルであることを特徴とする、請求項17に記載のプログラム記憶装置。

【請求項19】前記システム・コールがspawn()システム・コールであることを特徴とする、請求項17に記載のプログラム記憶装置。

【請求項20】前記ユーザがユーザIDを有し、前記アイデンティティが前記ユーザIDを含むことを特徴とする、請求項17に記載のプログラム記憶装置。

【請求項21】前記環境変数が前記ユーザIDと等しく設定されることを特徴とする、請求項20に記載のプログラム記憶装置。

【請求項22】前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザIDを有し、前記セキュリティ環境を作成する前記ステップ(b)

[2] が、前記新しいアドレス空間のユーザIDを、前記環境変数によって指定されたユーザ名のユーザIDと等しく設定するステップを含むことを特徴とする、請求項21に記載のプログラム記憶装置。

【請求項23】前記ユーザ名の前記ユーザIDを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、請求項22に記載のプログラム記憶装置。

【請求項24】前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループIDを有

し、

前記セキュリティ環境を作成する前記ステップ(b)

[2] が、前記新しいアドレス空間のグループIDを、前記環境変数によって指定されたユーザ名のグループIDと等しく設定するステップを含むことを特徴とする、請求項21に記載のプログラム記憶装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、新しいユーザ・アイデンティティと適切な特権を使用してPOSIX環境に新しい作業単位を作成する方法に関する。

【0002】

【従来の技術】(複数の相互接続された機械に作業が分散される)分散コンピューティング・システムは、クライアント/サーバ・モデルに基づいて構築されることが多い。このモデルでは、クライアント・プロセス(または単に「クライアント」)がサーバ・プロセス(または単に「サーバ」)に対して、プリント・サーバの場合はファイルの印刷、ファイル・サーバの場合はファイルの取り出しまたは記憶、あるいはアプリケーション・サーバの場合はアプリケーションの実行など、指定したサービスを実行するように求める要求を出す。クライアント・プロセスとサーバ・プロセスとは同じ物理機械上に存在することができるが、両者は異なる機械上に存在するのが一般的であり、以下の説明においてもそうである。サーバはローカル・エリア・ネットワーク(LAN)などのほか、インターネットなどのより広域のネットワークでも使用される。本明細書で特に扱うインターネット・サーバの1つのクラスは、ワールド・ワイド・ウェブにサービスを提供するサーバである。ワールド・ワイド・ウェブは、グラフィカル・コンテンツ(「ウェブ・ページ」)を提供し、ハイパーテキスト転送プロトコル(HTTP)に従ってその他のサービスを行うインターネット・サイトの集まりである。

【0003】ワールド・ワイド・ウェブ(または単に「ウェブ」)上のサーバなどのサーバは、オペレーティング・システムと共に機能し、オペレーティング・システムはサーバが存在する機械上でシステム資源を管理し、基本システム・サービスを行う。これらのオペレーティング・システムは、UNIXオペレーティング・システムまたはUNIXベースのオペレーティング・システムであることが多い。様々なUNIXオペレーティング・システムの急増により、そのようなシステムによって提供される共通のサービスのセットを定義する様々な努力がなされてきた。このような1つの努力は、1988年に初めて発表されたIEEE POSIX 1003.1仕様と、IEEE POSIX 1003.1dなどの補遺である(以下まとめて「POSIX」と呼ぶ)。

【0004】POSIX準拠システムでは、ユーザによ

10

20

30

40

50

るファイルなどへのアクセスは、各プロセス及び各ファイルに、ユーザのアイデンティティを指定するユーザIDとグループIDを関連づけることによって制御する。各ファイルには、3つの3ビット・フィールドも関連づけられ、各フィールドはrwxビットと呼ばれる3ビットの許可ビットから成る。この3つのフィールドはそれぞれ、ファイルの所有者と、所有者のグループ内の他のユーザと、所有者のグループに入っていないその他のユーザとに対して許可されたアクセスを定義する。各3ビット・フィールド内で、rビットはファイルの読み取りができるかどうかを指定し、wビットはファイルの書き込みができるかどうかを指定し、xビットはファイルの実行ができるかどうかを指定する。プロセスがファイルへのアクセスを要求するときは必ず、プロセスとファイルのユーザIDとグループIDと、そのファイルのために指定された許可ビットを検査して、そのプロセスを実行しているユーザが要求するアクセスを行うことができるかどうか判断される。このアクセス制御手続きは、当技術分野で周知であり、W. R. スティーブンス (Stevens) の「UNIX Network Programming」(1990年) 及びA. S. タネンバウム (Tanenbaum) の「Modern Operating System」(1992年) などの参考資料に記載されている。

【0005】サーバ環境でこのようなアクセス制御を実現するために、POSIX準拠オペレーティング・システムは、新しいユーザ・アイデンティティを使用して新しい作業単位と、その新しい作業単位が異なる1組の資源にアクセスすることができる新しいセキュリティ環境とを作成することができるサーバの能力をサポートしなければならない。このタイプの作業は、従来、サーバ・システム内で「デーモン」と呼ばれる背景プログラムによって行われてきた。一般に、デーモンは、デーモンの権限のサブセットを有するユーザに代わって要求を処理し、タスクを実行する。システムのセキュリティを維持するために、タスクはそのユーザの権限を使用して新しい作業単位の下で実行されなければならない。これらの新しい作業単位の作成は、fork()、setuid()、setgid()、exec()などの関数及びその他の関連するPOSIXサービスを使用して扱われる。

【0006】デーモンの例は、POSIXシステム上のウェブ・サーバである。このようなウェブ・サーバは、ポートでユーザがPOSIXファイル・システムから文書を取り出す要求を渡すのを監視するプログラムである。これを行うために、ウェブ・サーバはまずユーザ名とパスワードを検証しなければならない。次に、ウェブ・サーバはfork()関数を使用して新しいアドレス空間を作成し、要求されたユーザ・タスクのために別個の環境を設ける。新しいアドレス空間に入ると、ウェブ

サーバはgetgroups()やsetgroups()などの様々なPOSIX関数を使用して正しいセキュリティ環境を作成し、正しい補足グループIDを設定した後、setgid()及びsetuid()を使用して正しいグループIDとユーザIDを設定しなければならない。補足グループとグループIDとユーザIDとは、POSIX許可の基礎を形成する。最後にウェブ・サーバはシェルのexec()を行い、要求された文書を獲得するシェル・スクリプトを実行する。全体的に見て、これは大量の処理オーバーヘッドを要する長くて複雑なプロセスである。

【0007】

【発明が解決しようとする課題】spawn()サービスは、fork()サービスとexec()サービスを結合して1つのコールにするPOSIX関数である(IEEE POSIX 1003.1dに記載されている)。したがって、POSIXアプリケーションは、新しいアドレス空間をフォークしてから新しいプログラム・イメージを実行する代わりに、単にその新しいプログラムを異なるアドレス空間に作成して、要求側のアドレス空間をコピーするオーバーヘッドを節約することができる。しかし、この場合の問題は、現在のところ、デーモン自体のセキュリティ環境を変更せずに、新しいプログラム・イメージが制御を獲得する前にユーザ・アイデンティティを変更する方法がないことである。したがって、デーモンが要求された文書を獲得するためにシェル・スクリプトを作成しようとするれば、デーモンのユーザ・アイデンティティの下でPOSIXファイル・システムへのアクセスが行われることになり、それによって文書を要求しているユーザのPOSIX許可が置き換えられる可能性がある。spawn()を使用することができないため、デーモン・アプリケーションは前述のように、ユーザ・タスク・プロセス層で実行する余分の面倒なソフトウェアを備えなければならない。

【0008】

【課題を解決するための手段】一般には、本発明は、指定したユーザ・タスクを実行するように求めるユーザからの要求をデーモン・プロセスが監視するサーバ・システムにおいて、ユーザのための適切なセキュリティ環境を使用してユーザに代わってタスクを実行する方法及び装置に関する。デーモン・プロセスは、ユーザから要求を受け取ると、その要求で指定されているユーザ・アイデンティティ(たとえばユーザ名)に従って環境変数を設定し、オペレーティング・システム・カーネルに対して、指定されたユーザ・タスクを新しいアドレス空間で実行するシステム・コールを出す。オペレーティング・システム・カーネルは、POSIX準拠カーネルとすることができ、その場合、システム・コールはspawn()システム・コールとすることができ、オペレーティング・システム・カーネルはデーモン・プロセスから

システム・コールを受け取ると、指定されたユーザ・タスクのために新しいアドレス空間を作成し、指定されたユーザ・タスクのために環境変数に従って新しいセキュリティ環境を作成し、新しいアドレス空間内で指定されたユーザ・タスクを実行する。

【0009】具体的には、本発明によると、新しい環境変数 (USERNAME) を作成することによって従来の技術の前述の欠点が解消される。この環境変数が指定されると、spawn () 関数によって呼び出されるプログラムが正しいPOSIX許可を使用して制御を獲得する。これを実現するために、spawn () 関数を変更して、spawn () 関数がこの新しい環境変数を認識し、有効なユーザ名であることを検証した後、指定されたユーザ名のために新しいアドレス空間の初期作成が行われるようにする。新しいユーザ・アイデンティティを使用して実行されると、セキュリティ・データベースに入っているそのユーザのエンティティから残りのPOSIX許可を入手する。新しいアドレス空間とタスクの初期設定が完了すると、spawnルーチンはその新しいプログラム・イメージを現行セキュリティ環境で開始する。

【0010】本発明は、デーモン・プロセス環境に変更を加える必要なしに、ユーザ・タスク・プロセスのためのセキュリティ環境を作成する方法を提供する。さらに、デーモン・アプリケーションのどの部分もユーザ・タスク・プロセスで実行する必要がない (もしそうであればユーザ・タスク・プログラム・イメージに制御を渡す前にユーザ・タスク・プロセス許可の変更を必要とすることになる)。また、本発明は、従来のfork () とexec () の使用をspawn () 関数の使用で置き換えることができ、それによってspawn () 関数が本質的にもたらす簡略化とパフォーマンス向上を実現する。

【0011】

【発明の実施の形態】図1は、従来の実施態様のデーモンを組み込んだコンピュータ・システム100 (物理機械を含む) の例であり、特定のユーザのために要求されたタスクを実行する様々なソフトウェア層間の関係が示されている。これらの層には、デーモン・プロセス層またはアドレス空間102と、オペレーティング・システム (OS) またはカーネル層またはアドレス空間104と、後述のようにして作成されるユーザ・タスク・プロセス層またはアドレス空間106とが含まれる。物理機械は、たとえばS/390並列エンタープライズ・サーバなどのIBM S/390プロセッサとすることができ、カーネル層104はPOSIX準拠OpenEdition構成要素を有するIBM OS/390オペレーティング・システムとすることができる。

【0012】デーモン・プロセス層102には、ユーザに代わって動作して遠隔クライアント (図示せず) に結

合されたポートまたは通信回線110を監視するソフトウェアを含むデーモン・アプリケーション108がある。デーモン・アプリケーション108は、入力としてユーザ名114とパスワード116とタスクを指定する識別子118とを含む要求112をポート110を介して受け入れる (ステップ120)。

【0013】ユーザ名 (またはログイン名) 114は、ユーザに固有に関連づけられ、ユーザが自分自身をシステム100に識別させるために使用する英数字文字列である。システム100に付随するセキュリティ・データベース134内の各ユーザ名114のレコード136に、ユーザ名に固有に関連づけられた整数ユーザIDと、ユーザ名に固有に関連づけられた整数グループIDと、ユーザ名に関連づけられた補足グループのリストのほか、ユーザのパスワード及びその他の関係情報が記憶される。ユーザを認証するために、デーモン・アプリケーション108はセキュリティ・データベース134内の指定されたユーザ名114に対応するレコード136にアクセスし、そのレコードに入っているパスワードを調べる。指定されたユーザ名114のレコード136があり、レコード内のパスワードが要求に添付されたパスワード118と一致する場合、要求者はその要求者が自称しているユーザであると認証される。

【0014】デーモン・アプリケーション108はユーザ名114を検証し、パスワード116を認証した後、要求されたタスクを実行する新しいプログラム・イメージを実行するように新しい環境を準備しなければならない。そのために、デーモン・アプリケーション108はカーネル層104にfork () システム・コールを出して新しいプロセスを作成する (ステップ122)。

【0015】fork () システム・コールは、カーネル層104でカーネルforkルーチン124をトリガして制御を獲得させる。forkルーチン124は制御を獲得すると新しいアドレス空間106を作成して初期設定し (ステップ126)、デーモン層102からユーザ・タスク・プロセス層106に記憶属性、セキュリティ属性、及びプロセス実行属性をコピーする (ステップ128)。

【0016】正常に完了すると、fork () ルーチン124はデーモン・アプリケーション108を持つプロセス層102が、デーモン・アプリケーション130を持つ新たに作成された子プロセス層106の親になるようにする。fork () ルーチン124から、デーモン・アプリケーション108及び130にそれらのデーモン・アプリケーションがどのプロセス層で実行されているかを示す標識を返す。親プロセス層102内の場合、その層内のデーモン・アプリケーション108はループ・バックして、ポート110からさらに作業が送られてくるのを待つ (ステップ129)。

【0017】子層106内の場合、その層内のデーモン

10

20

30

40

50

・アプリケーション114は要求で指定されたユーザ名114のセキュリティ属性を設定する。

【0018】いくつかのPOSIX関数が呼び出され、子層106の正しい補足グループとグループIDとユーザIDとが設定される。これらのコールは、セキュリティ・データベース134内の要求側ユーザのデータにアクセスする。具体的には、子デーモン・アプリケーション130はまずgetpwnam() コールを出してセキュリティ・データベース134にアクセスし、ユーザ名114に対応するユーザIDとグループIDを判断する(ステップ132)。アプリケーション130は次にgetgroups() コールを出してセキュリティ・データベース134にアクセスし、ユーザ名114に対応する補足グループを判断する(ステップ138)。

【0019】この情報を使用して、子デーモン・アプリケーション130はsetgroups() コールを出して子層106の補足グループをユーザ名114に対応する補足グループに設定し(ステップ140)、setgid() コールを出して子層106のグループIDをユーザ名114に対応するグループIDに設定し(ステップ142)、setuid() コールを出して子層106のユーザIDをユーザ名に対応するユーザIDに設定する(ステップ144)。

【0020】上述のようにして新しいユーザのために正しいPOSIXセキュリティ環境が設定されると、デーモン・アプリケーション114は指定されたユーザ・タスク識別子118をパラメータとして使用してexec() システム・コールを出す(ステップ146)。この関数によってカーネルexecルーチン148はアドレス空間106を初期設定し直し、記憶域を消去し、正しいプロセス・セキュリティと実行環境を設定し(ステップ150)、その後で新しいユーザ・タスク・プログラム154に制御を渡す(ステップ152)。

【0021】OpenEdition拡張機能付きのIBM OS/390オペレーティング・システムなどの多くのPOSIX準拠システムでは、ユーザ・プロセス内のユーザ・アイデンティティを変更するためにセキュリティ・データベースに対して行われる過度のコールに関する重大なパフォーマンス上の不利がある。また、この従来の方法では、デーモンはPOSIXのspawn() サービスを利用することができず、したがってデーモン・アプリケーションの一部を子プロセスで実行するようにななければならない。

【0022】図2は、本発明を組み込んだコンピュータ・システム200(物理機械を含む)の例を示す図であり、指定されたユーザIDのために要求されたタスクを実行する様々なソフトウェア層間の関係が図示されている。これらの層には、デーモン・プロセス層またはアドレス空間202と、オペレーティング・システム(OS)またはカーネル層またはアドレス空間204と、

(作成された場合は)ユーザ・タスク・プロセス層またはアドレス空間206とが含まれる。

【0023】デーモン・プロセス層202には、遠隔クライアント(図示せず)に結合されたポートまたは通信回線210を監視するソフトウェアを含むデーモン・アプリケーション208がある。デーモン・アプリケーション208は、入力としてユーザID214とパスワード216と実行する特定のタスクの識別子218とを含むクライアント要求212をポート210を介して受け入れる(ステップ220)。

【0024】デーモン・アプリケーション208は、ユーザID214を検証し、パスワード216を認証した後、要求されたタスクを行う新しいプログラム・イメージを実行するように新しい環境を準備する。これを行うために、デーモン・アプリケーション208はまず、環境変数USERNAMEを処理する要求212のユーザ名214に設定する(ステップ222)。デーモン・アプリケーション208は次に、カーネル層204に対してspawn() システム・コールを出して新しいプロセスを作成する(ステップ224)。spawn() システム・コールはパラメータとしてタスク識別子218を渡し、環境変数USERNAMEとしてユーザ名214を渡す。次にデーモン・アプリケーション208はルーチンの先頭にループ・バックしてさらに作業を入手する(ステップ236)。

【0025】spawn() システム・コール(ステップ224)によって、カーネル層204内のカーネルspawnルーチン226が制御を獲得する。spawnルーチン226は制御を獲得するとまず新しいユーザ・タスク層またはアドレス空間206を作成する(ステップ228)。

【0026】spawnルーチン226は次に、カーネル層204内のセキュリティ・データベース232に照会して必要なPOSIX許可設定値を入手する(ステップ230)。次に、正しいセキュリティ許可を使用して、すなわちデータベース232内の(環境変数USERNAMEに指定されている)ユーザ名214に対応づけられたグループIDとユーザIDに等しく設定された新しいアドレス空間206のユーザIDとグループIDを使用して、新しいアドレス空間206を直接初期設定する。

【0027】最後に、カーネルspawnルーチン226は、要求212で指定された(デーモン・アプリケーション208によってパラメータとして渡された)ユーザ・タスク218に対応するプログラム・イメージ238をアドレス空間206にロードし、そのプログラム・イメージに制御を渡してユーザ・タスクを実行する(ステップ234)。

【0028】遠隔ユーザによる悪用を避けるために、新しい環境変数USERNAMEは許可されたユーザのみ

10

20

30

40

50

に制限する必要がある。この機能を使用するのに必要な特権は、`sctuid()` 関数と同等でなければならない。(デーモン・アプリケーションは通常そうであるように)デーモン・アプリケーション208が無制限のアクセス権を持つスーパーユーザ(ユーザID=0)として動作する場合、デーモン・アプリケーション208は必要な権限を持つことになる。

【0029】上述のように、本発明は、デーモン・プロセス環境に変更を加える必要なしに、また、デーモン・アプリケーションのどの部分もユーザ・タスク・プロセス内で実行する必要なしに、ユーザ・タスク・プロセスのためのセキュリティ環境を作成する方法を提供する。また、本発明は、従来の`fork()` 関数と`exec()` 関数を`spawn()` 関数で置き換えることができるようにし、それによって`spawn()` 関数が本質的に備える簡略さとパフォーマンス強化を実現する。

【0030】当業者なら様々な修正がわかるであろう。上記のように本発明についてUNIXベースのシステム、具体的にはPOSIX準拠システムの文脈で説明したが、本発明はその他のシステムでも使用可能であることは明らかであろう。

【0031】まとめとして、本発明の構成に関して以下の事項を開示する。

【0032】(1) 要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視するサーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する方法であって、

(a) デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、[1] 前記要求で指定された前記アイデンティティに従って環境変数を設定するステップと、[2] 前記オペレーティング・システム・カーネルに対してシステム・コールを出して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

(b) 前記オペレーティング・システム・カーネルが前記デーモン・プロセスから前記システム・コールを受け取ると、[1] 前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、[2] 前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、[3] 前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始するステップとを含む方法。

(2) 前記オペレーティング・システム・カーネルがPOSIX準拠オペレーティング・システム・カーネルであることを特徴とする、上記(1)に記載の方法。

(3) 前記システム・コールが`spawn()` システム・コールであることを特徴とする、上記(1)に記載の方法。

(4) 前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、上記(1)に記載の方法。

(5) 前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、上記(4)に記載の方法。

(6) 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザIDを有し、前記セキュリティ環境を作成する前記ステップ(b) [2] が、前記新しいユーザ・アドレス空間のユーザIDを、前記環境変数によって指定されたユーザ名のユーザIDと等しく設定することを特徴とする、上記(5)に記載の方法。

(7) 前記ユーザ名のユーザIDを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、上記(6)に記載の方法。

(8) 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループIDを有し、前記セキュリティ環境を作成する前記ステップ(b) [2] が、前記新しいアドレス空間のグループIDを、前記環境変数によって指定されたユーザ名のグループIDと等しく設定することを特徴とする、上記(5)に記載の方法。

(9) 要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視するサーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わって前記タスクを実行する装置であって、

(a) 前記デーモン・プロセスに関連づけられ、前記ユーザからの前記要求の受取りにตอบสนองして、[1] 前記要求に指定された前記アイデンティティに従って環境変数を設定し、[2] 前記オペレーティング・システム・カーネルに対してシステム・コールを出して新しいアドレス空間内で前記指定されたユーザ・タスクを実行する手段と、

(b) 前記オペレーティング・システム・カーネルに関連づけられ、前記デーモン・プロセスからの前記システム・コールの受取りにตอบสนองして、[1] 前記指定されたユーザ・タスクのために新しいアドレス空間を作成し、[2] 前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成し、[3] 前記新しいアドレス空間内で前記指定されたユーザ・タスクを開始する手段とを含む装置。

(10) 前記オペレーティング・システム・カーネルがPOSIX準拠オペレーティング・システム・カーネルであることを特徴とする、上記(9)に記載の装置。

(11) 前記システム・コールが`spawn()` システム・コールであることを特徴とする、上記(9)に記載の装置。



(12) 前記ユーザがユーザ名を有し、前記アイデンティティが前記ユーザ名を含むことを特徴とする、上記(9)に記載の装置。

(13) 前記環境変数が前記ユーザ名と等しく設定されることを特徴とする、上記(12)に記載の装置。

(14) 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザIDを有し、前記セキュリティ環境を作成する前記手段(b) [2] が、前記新しいアドレス空間のユーザIDを、前記環境変数によって指定されたユーザ名のユーザIDに等しく設定する手段を含むことを特徴とする、上記(13)に記載の装置。

(15) 前記ユーザ名のユーザIDを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、上記(14)に記載の装置。

(16) 前記アドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループIDを有し、前記セキュリティ環境を作成する前記ステップ(b) [2] が、前記新しいアドレス空間のグループIDを、前記環境変数によって指定されたユーザ名のグループIDに等しく設定することを特徴とする、上記(13)に記載の装置。

(17) 要求がユーザのアイデンティティを指定し、サーバ・システムがオペレーティング・システム・カーネルを有する、指定されたユーザ・タスクの実行を求める前記ユーザからの前記要求をデーモン・プロセスが監視する前記サーバ・システムにおいて、前記ユーザのための適切なセキュリティ環境を使用して前記ユーザに代わってタスクを実行する方法ステップを行う機械によって実行可能なプログラムを有形に実施する機械可読プログラム記憶装置であって、前記方法ステップが、

(a) 前記デーモン・プロセスが、前記ユーザからの前記要求を受け取ると、[1] 前記要求に指定された前記アイデンティティに従って環境変数を設定するステップと、[2] 前記オペレーティング・システムに対してシステム・コールを発行して前記指定されたユーザ・タスクを新しいアドレス空間内で実行するステップと、

(b) 前記オペレーティング・システム・カーネルが、前記デーモン・プロセスから前記システム・コールを受け取ると、[1] 前記指定されたユーザ・タスクのために新しいアドレス空間を作成するステップと、[2] 前記環境変数に従って前記指定されたユーザ・タスクのためにセキュリティ環境を作成するステップと、[3] 前記指定されたユーザ・タスクを前記新しいアドレス空間内で開始するステップとを含む、プログラム記憶装置。

(18) 前記オペレーティング・システム・カーネルがPOSIX準拠オペレーティング・システム・カーネルであることを特徴とする、上記(17)に記載のプログラ

ム記憶装置。

(19) 前記システム・コールがspawn() システム・コールであることを特徴とする、上記(17)に記載のプログラム記憶装置。

(20) 前記ユーザがユーザIDを有し、前記アイデンティティが前記ユーザIDを含むことを特徴とする、上記(17)に記載のプログラム記憶装置。

(21) 前記環境変数が前記ユーザIDと等しく設定されることを特徴とする、上記(20)に記載のプログラム記憶装置。

(22) 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたユーザIDを有し、前記セキュリティ環境を作成する前記ステップ(b) [2] が、前記新しいアドレス空間のユーザIDを、前記環境変数によって指定されたユーザ名のユーザIDと等しく設定するステップを含むことを特徴とする、上記(21)に記載のプログラム記憶装置。

(23) 前記ユーザ名の前記ユーザIDを、セキュリティ・データベースにアクセスすることによって判断することを特徴とする、上記(22)に記載のプログラム記憶装置。

(24) 前記新しいアドレス空間と前記ユーザ名とがそれぞれそれに関連づけられたグループIDを有し、前記セキュリティ環境を作成する前記ステップ(b) [2] が、前記新しいアドレス空間のグループIDを、前記環境変数によって指定されたユーザ名のグループIDと等しく設定するステップを含むことを特徴とする、上記(21)に記載のプログラム記憶装置。

【図面の簡単な説明】

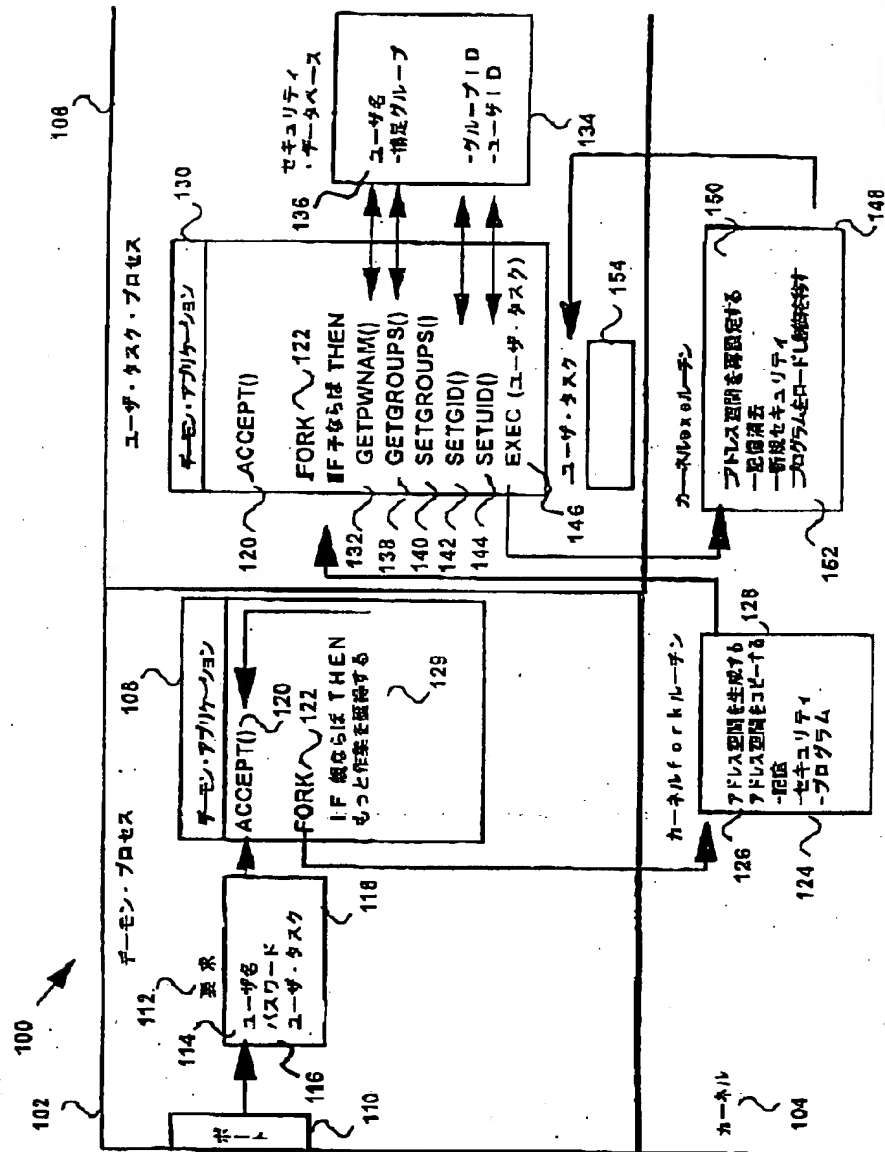
【図1】 ユーザ要求タスクのためのセキュリティ環境を作成する従来の実施態様を組み込んだコンピュータ・システムを示す図である。

【図2】 ユーザ要求タスクのためのセキュリティ環境を作成する本発明の組込みを使用したコンピュータ・システムを示す図である。

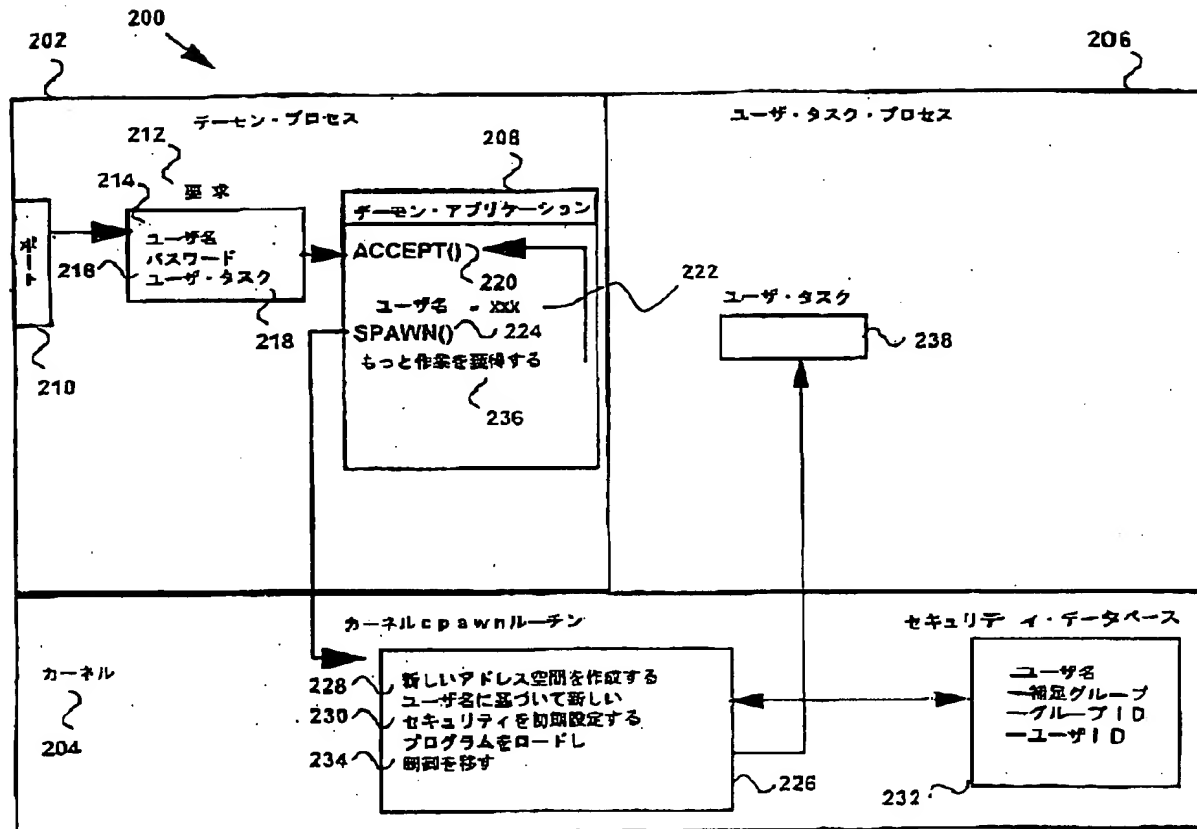
【符号の説明】

200 コンピュータ・システム  
202 デーモン・プロセス層  
204 カーネル層  
206 ユーザ・タスク・プロセス層  
208 デーモン・アプリケーション  
210 ポート  
214 ユーザ名  
226 カーネルspawnルーチン  
232 セキュリティ・データベース  
238 ユーザ・タスク

【図1】



【図2】



フロントページの続き

(72) 発明者 アーネスト・スコット・ペンダー  
 アメリカ合衆国12477 ニューヨーク州ソ  
 ージャーティーズ バイン・グローブ・ス  
 クール・ロード 27

(72) 発明者 マイケル・ゲイリー・スピーゲル  
 アメリカ合衆国10950 ニューヨーク州モ  
 ンローサンセット・ハイツ 10